Part1 Design Document

13302010017 Guo Liangchen

Find the opcode

I find the two opcodes in the same way. For example, `test1`. Just the following steps: 1. Disassemble the binary files given:

```
sslittle-na-sstrix-objdump -x -d test1 > test1.asm
```

2. Run the `sim-fast` with gdb:

```
gdb sim-fast
```

3. Set up the breakpoint, then run the simulator:

```
(gdb) b sim-fast.c : 444
Breakpoint 1 at 0x40e377: file sim-fast.c, line 444.
(gdb) r test1
```

4. Get the `inst` by using gdb's print command:

```
(gdb) p/x inst
```

Then we get the `inst` is `0x10111300` 5. Just find the `inst` in the file `test1.asm` which we just generated, we will get:

```
...
00400270 <addOK+30> 0x00000061:10111300
...
```

Then we find the opcode of `test1` is `0x61`.

Add the instruction

addok

The solution is simple. Just copy the `add` instruction definition and modify it: 1. Modify the `opcode` to `0x61` and the name of the instruction. 2. Modify the implemetation to:

```
if (OVER(GPR(RS), GPR(RT)))
   SET_GPR(RD, 0);
else
   SET_GPR(RD, 1);
```

bitcount

The solution based on divide-and-conquer. This page describes the solution. 1. Modify the `opcode` to `0x62` and the name of the instruction. 2. Modify the implemetation to the following:

```
unsigned int v = GPR(RS);
unsigned int c;
c = v - ((v >> 1) & 0x55555555);
c = ((c >> 2) & 0x33333333) + (c & 0x33333333);
c = ((c >> 4) + c) & 0x0F0F0F0F;
c = ((c >> 8) + c) & 0x00FF00FF;
c = ((c >> 16) + c) & 0x0000FFFF;
if (UIMM == 0)
    SET_GPR(RT, 32 - c);
else
    SET_GPR(RT, c);
```